



# Kuidas saada (heaks) programmeerijaks?

Margus Freudenthal

[margus@cyber.ee](mailto:margus@cyber.ee)

# Minu taust

- ⊙ Alates 1999. aastast töotan Cyberneticas
  - ⊙ Erakapitalil põhinev Eesti ettevõte
  - ⊙ 170 töötajat, tugev rõhk teadusel
  - ⊙ Palju keerulisi süsteeme, palju e-riiki
- ⊙ Olen teinud küllaltki erinevaid asju
  - ⊙ Digitaalalkirja süsteemid, ajatembeldamine, X-tee
  - ⊙ Mereraudiosüsteem
  - ⊙ Mitmed „tavalised” infosüsteemid, suurim neist Eesti Maksu- ja Tolliametis
  - ⊙ PhD – valdkonnaspetsiifiliste keelte arendus
- ⊙ Hetkel e-riigi toote (UXP) peaarhitekt ja arendusjuht

# Põhiline

*Vita brevis, ars longa, occasio praeceps, experimentum periculosum, iudicium difficile*

- ⊙ Mingil alal meisterlikkuse saavutamiseks kulub 10 aastat
  - ⊙ Ka Mozartil
- ⊙ See 10 aastat peab olema täidetud sihikindla tööga õppimist soodustavas keskkonnas
- ⊙ Teadmised ja oskused on otseses sõltuvuses neisse investeeritud ajast

# Harjutamine

- ⊙ Tööta koos tugevamatega
- ⊙ Õpeta teisi
- ⊙ Mõtle, kuidas saaks järgmine kord paremini teha
- ⊙ Nokitse oma hobiprojektide kallal
- ⊙ Ütle väljakutsetele „jah”
- ⊙ Töötundide panustamine on lihtsam, kui ala tõeliselt meeldib
  - ⊙ Leia enda jaoks motivaator: pidev areng, maailma parandamine, ...

# Mida õppida?

- ⊙ Tänapäeva maailm on kiire
- ⊙ Õpitu vananeb paari aastaga

# Mida õppida?

- ⊙ Tänapäeva maailm on kiire
- ⊙ Õpitu vananeb paari aastaga

## Ära õpi vananevaid asju!

# Programmeerimisoskus ei vanane

# Nagu näiteks?

- ⊙ Knuthi „The art of computer programming”
  - ⊙ 1968-73
- ⊙ Objektorienteeritud programmeerimine
  - ⊙ 1970. algus
- ⊙ Funktsionaalprogrammeerimine
  - ⊙ 1950. lõpp
- ⊙ Enamik moodsaid paradigmasid
  - ⊙ Teenusorienteeritud arhitektuur – seitsmekümnendad
  - ⊙ *Reactive programming* – kuuekümnendad (*dataflow programming*)



# Aga...

- ⊙ Konkreetseid, tänapäevaseid vahendeid on vaja osata
- ⊙ Säilita õige perspektiiv
- ⊙ Vahend on vahend
- ⊙ Programmeerija oled ikkagi sina

# Spetsialiseerumisest

*Only by strict specialization can the scientific worker become fully conscious, for once and perhaps never again in his lifetime, that he has achieved something that will endure. A really definitive and good accomplishment is today always a specialized act.*

-- Max Weber

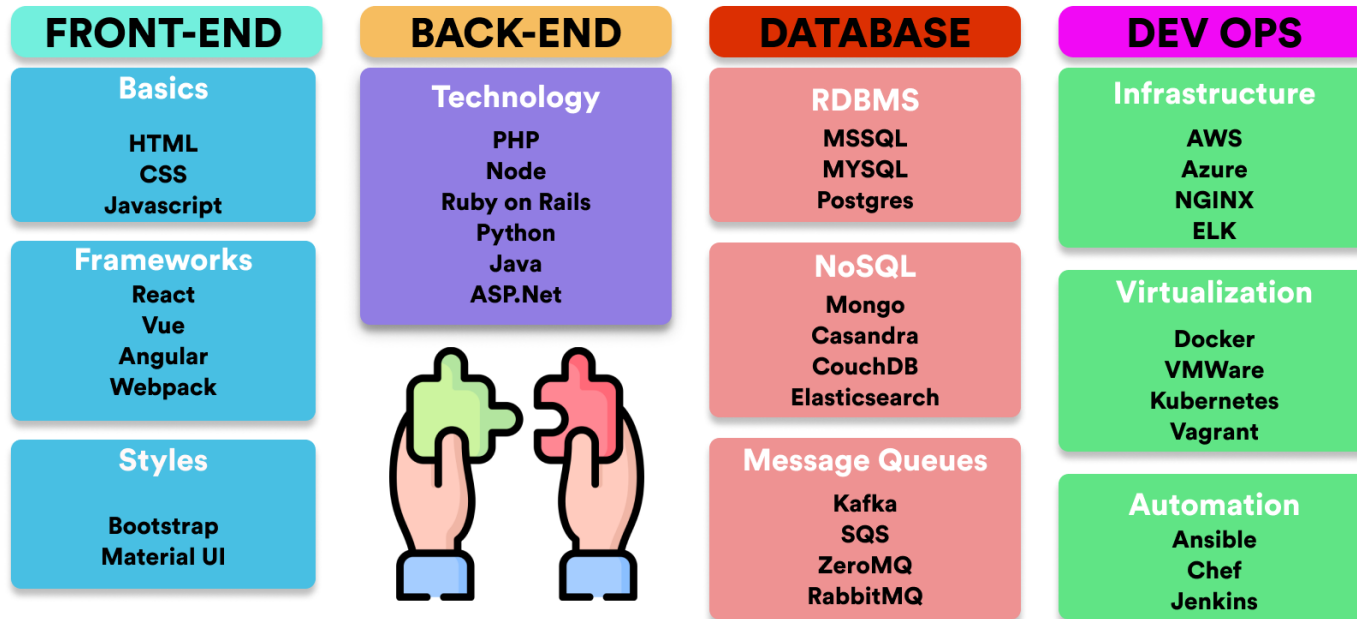
# Spetsialiseerumisest

*A human being should be able to change a diaper, plan an invasion, butcher a hog, conn a ship, design a building, write a sonnet, balance accounts, build a wall, set a bone, comfort the dying, take orders, give orders, cooperate, act alone, solve equations, analyze a new problem, pitch manure, program a computer, cook a tasty meal, fight efficiently, die gallantly. Specialization is for insects.*

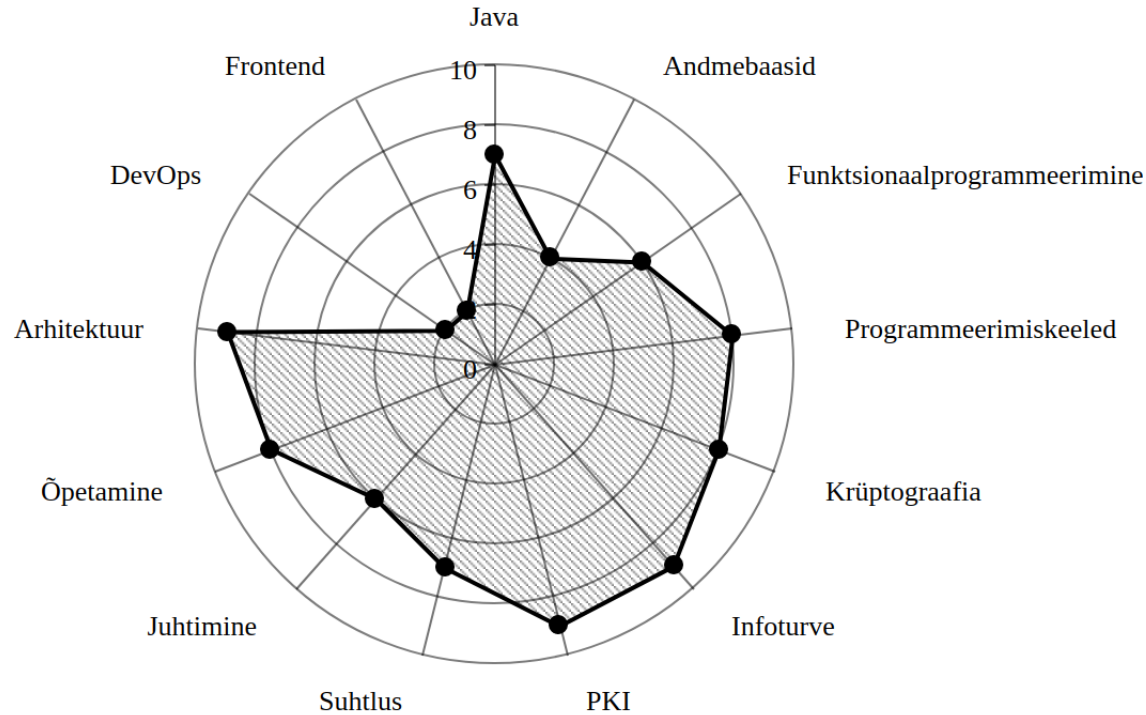
-- Robert A. Heinlein

# Spetsialiseerumisest

## Full-Stack Developer Skills Breakdown



# Spetsialiseerumisest



# Programmeerimisoskus

- ⊙ Programmeerimine ei ole tegelikult ühtne oskus
- ⊙ Suure pildi nägemine, komponendid, protokollid, hajussüsteemid
- ⊙ Algoritmid, optimeerimine
- ⊙ Koodi struktureerimine, ilumeel
- ⊙ Koodi lugemine, tõrkeanalüüs (*debugging*), testimine
- ⊙ Konkreetsed keeled, teegid, raamistikud
- ⊙ Tiimitöö, inimestega suhtlemine, juhendamine, juhtimine

# Kaks suurt valdkonda

- ⊙ Arvutiteadus (*computer science*)
  - ⊙ Algoritmid
  - ⊙ Matemaatika
  - ⊙ Teadus
- ⊙ Tarkvaratehnika (*software engineering*)
  - ⊙ Kuidas tarkvara arendada, tööprotsessid
  - ⊙ Kuidas koodi struktureerida
  - ⊙ Milline on hea arhitektuur
  - ⊙ Suuresti intuitiivne

# Kirjanduse paradoks

- ⊙ Tarkvara projekteerimise alaste raamatute lugemine ei ole võluvits
- ⊙ Nende sisust saad aru alles siis, kui sa asja juba oskad
- ⊙ Tuleb lugeda mitu korda
- ⊙ Hiljem annavad raamistiku ja terminoloogia
  - ⊙ Kasulik mõtlemiseks ja suhtlemiseks

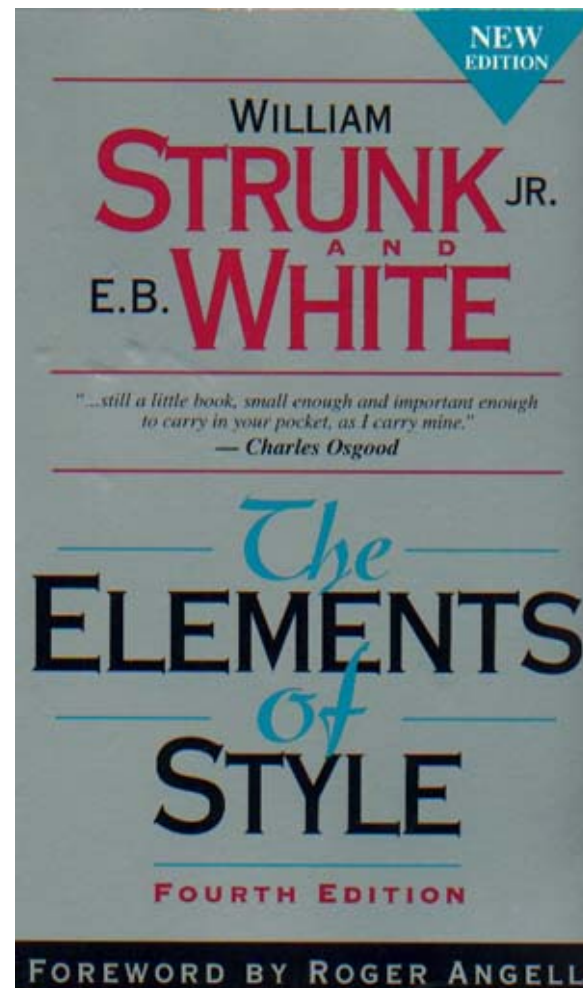


# Täpsusest

- ⊙ Programmeerimine seisneb asjaolude täpses ja detailses üleskirjutamises
- ⊙ Pedantsus on oluline isikuomadus
- ⊙ Programmeeri täpselt
- ⊙ Mõtle täpselt
- ⊙ Räägi täpselt

# Õpi inimkeelt

- ◎ Õpi korrektselt kirjutama eesti ja inglise keeles



# Õpi programmeerimiskeeli

- ⊙ Need on programmeerija põhilised tööriistad
- ⊙ Sealt saab praktilisi oskusi
- ⊙ Sealt saab laiemat silmaringi
  - ⊙ Uued mõisted
  - ⊙ Uued mustrid
  - ⊙ Uued mõttemallid
- ⊙ Esinevad ka muudes keeltes, aga sageli ilmutamata kujul

# Õpi programmeerimiskeeli

- ⊙ Programmeerimiskeele reaalseks omandamiseks kulub paar aastat intensiivset kasutamist
- ⊙ Samas, programmeerimiskeelte oskus ei vanane kuigi kiiresti
- ⊙ Mõisted ja paradigmad on veel püsivamad

# Mida õppida?

- ⊙ C
  - ⊙ Platvormist sõltumatu assembler
- ⊙ Ruby, Smalltalk (Python?)
  - ⊙ Objektorienteeritud programmeerimine
- ⊙ JavaScript/TypeScript
  - ⊙ Sest meil on aasta 2020... Lisaks huvitav objektimudel
- ⊙ Java, C#
  - ⊙ 21. sajandi COBOL, praktikas vajalik
- ⊙ Python, Ruby
  - ⊙ Lihtsad asjad kiirelt valmis

# Mida õppida?

- ⊙ Scheme, Haskell, OCaml (Scala?)
  - ⊙ Funktsionaalprogrammeerimine
  - ⊙ Kõrge abstraktsioonitasemega programmeerimine
- ⊙ Prolog
  - ⊙ Loogiline programmeerimine
- ⊙ Erlang
  - ⊙ Paralleelpprogrammeerimine

# Mine süvitsi

- ⊙ Ehita valmis programmeerimiskeele kompilaator või interpretaator
    - ⊙ Vali keel, mida tahad väga hästi tundma õppida
  - ⊙ Interpretaator on lihtsam
  - ⊙ Kompilaatoriga õpib rohkem
- 
- ⊙ Tulemus: uus viis, kuidas programmeerimiskeeli mõista

# Structure and Interpretation of Computer Programs

Second Edition



Harold Abelson and  
Gerald Jay Sussman  
with Julie Sussman



# Mida tasub teha

- ⊙ Tööta koos tugevatega
  - ⊙ Midagi jääb sulle ka külge
  - ⊙ Saad kriitikat ja tagasisidet
  - ⊙ Motiveerib pingutama
- ⊙ Harjuta
  - ⊙ Tee tööd mõttega
  - ⊙ Mõtle, kuidas saaks paremini
  - ⊙ Tee järgmine kord paremini
- ⊙ Harjuta väljaspool tööd
  - ⊙ Nokitse oma projektide kallal

# Mida tasub teha

- ⊙ Keera asju nässu
  - ⊙ Vähesed suudavad õppida teiste vigadest
- ⊙ Loe teiste programme
  - ⊙ Loe häid programme
  - ⊙ Õpi nägema ilu
- ⊙ Võta väljakutsed vastu
  - ⊙ Kui sa pole mingit asja varem teinud, siis nüüd on just õige aeg seda proovida
  - ⊙ Tekib harjumus uusi asju õppida, ülesandesse sisse elada

# Mida tasub teha

- ⊙ Õpi meeskonnas töötama
  - ⊙ Suuri asju ei tehta üksi
  - ⊙ Alati pead arvestama teistega
  - ⊙ Ka juhina
  - ⊙ Sinu panus lahustub üldises tervikus

# Motivatsioon

- ⊙ Töötundide panustamine on lihtsam, kui ala tõeliselt meeldib
  - ⊙ Otsi ala, mis sulle meeldib. :)
- ⊙ Naudi täiuslikkuse poole püüdlemise protsessi
- ⊙ Naudi võimet teiste elu paremaks teha
  
- ⊙ See aitab saada üle igavatest ja tüütutest kohtadest

# Ülikoolist

- ⊙ **Kindlasti lõpeta ülikool!**
- ⊙ Hea punkt CVs
- ⊙ Saad kogemusi ja sõpru
- ⊙ Õpid asju, mida muidu ei viitsiks









# Ära kasuta loitse

- ⊙ Mida teeb see koodirida?
- ⊙ Mida teeb see välja kutsutav funktsioon?
- ⊙ Mis juhtub selle sõnumi saatmisel?
- ⊙ Mismoodi see automaatne mehhanism töötab?
- ⊙ Miks me kasutame arendamisel seda mustrit?

# Lõpetuseks

*The programmer, like the poet, works only slightly removed from pure thought-stuff. He builds castles in the air, from air, creating by exertion of the imagination. [...] The magic of myth and legend has come true in our time. One types the correct incantation on a keyboard, and a display screen comes to life, showing things that never were nor could be. [...]*

-- Frederick P. Brooks, The Mythical Man-Month



CYBERNETICA